1267924

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

January 01, 2005

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: *60/519,129*
FILING DATE: *November 12, 2003*
RELATED PCT APPLICATION NUMBER: *PCT/US04/37761*

Certified by

Jon W Dudas

Acting Under Secretary of Commerce
for Intellectual Property
and Acting Director of the U.S.
Patent and Trademark Office

13049 U.S.PTO

22582 U.S.PTO 60/519129 111203

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

Express Mail No. EU200162692US

	Docket Number	126709.600	Type a plus sign (+) inside this box →	+

INVENTOR(s)/APPLICANT(s)

.AST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
Iorton	James	A.	New Tripoli, Pennsylvania
Klein	Robert	C., Jr.	Macungie, Pennsylvania

TITLE OF THE INVENTION (280 characters max)

SELF-ROUTING, MESSAGE-BASED INTERCONNECT SYSTEM FOR ELECTRICAL DEVICES

CORRESPONDENCE ADDRESS

Firm ID 21269

Pepper Hamilton LLP
500 Grant Street
One Mellon Center, 50th Floor
Pittsburgh, PA 15219

ENCLOSED APPLICATION PARTS (check all that apply)

X] Specification (Number of pages) [21] [ ] CDs (number) [ ]
X] Drawings (Number of sheets) [4 within text] [X] Other (specify): Postcard, Certificate of Mailing, Fee Transmittal and Check

METHOD OF PAYMENT (check one)

X] Applicant(s) claim(s) small entity status. See 37 C.F.R. § 1.27. X] A check or money order is enclosed to cover the Provisional filing fees X] The Commissioner is hereby authorized to charge fees to Deposit Account Number: 50-0436	PROVISIONAL FILING FEE AMOUNT ($)	$80.00

he invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

x] No [ ] Yes

] Additional inventors are being named on separately numbered sheets attached hereto.

Respectfully submitted,

James M. Singer
Registration No. 45,111

Date: November 12, 2003

F: #162840 v1 (3HNC01!.DOC)

CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.10

APPLICANT:	JAMES A. HORTON, ET AL.
TITLE:	SELF-ROUTING, MESSAGE-BASED INTERCONNECT SYSTEM FOR ELECTRICAL DEVICES
SERIAL NO.:	NOT YET ASSIGNED
ATTORNEY REF:	126709.600
DATE OF DEPOSIT:	NOVEMBER 12, 2003
EXPRESS MAIL NO.	EV200162692US

I HEREBY CERTIFY THAT THIS PROVISIONAL PATENT APPLICATION IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE VIA UNITED STATES POST OFFICE EXPRESS MAIL UNDER 37 C.F.R. § 1.10 ON THE DATE INDICATED ABOVE AND IS ADDRESSED TO THE MAIL STOP PROVISIONAL APPLICATION, COMMISSIONER FOR PATENTS, P.O. BOX 1450, ALEXANDRIA, VA 22313-1450.

KATHLEEN PUJOL
(Typed or printed name of person mailing paper or fee)

Kathleen P
(Signature of person mailing paper or fee)

DOCUMENTS ENCLOSED:

1. New Provisional Application Cover Sheet (1 sheet);
2. Provisional Application (21 pages);
3. Formal Figures (4 within text)
4. Fee Transmittal and check in the amount of $80.00; and
5. Certificate of Mailing;
6. Postcard.

SELF-ROUTING, MESSAGE-BASED INTERCONNECT
SYSTEM FOR ELECTRICAL DEVICES

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to an integrated circuit interconnect method and apparatus and more specifically it relates to a self-routing, message-based interconnect system for electrical devices that provides a flexible, efficient, self-routing and dynamically-optimized means for connecting together functional elements within a semiconductor device.

Description of the Prior Art

It can be appreciated that integrated circuits (chips or ICs) with various types of interconnect have been in use for years. Typical integrated circuit interconnections can be grouped into three distinct classes of on-chip interconnect. The first class is conventional microprocessor and/or microcontroller bus-type architectures. The second is the programmable interconnect found in Field Programmable Gate Arrays (FPGAs) and other Programmable Logic Devices (PLDs). The third class of on-chip interconnect is the highly customized, hard-wired, device-specific wiring that is found in full-custom, Application Specific Integrated Circuits (ASICs).

The main problem with conventional integrated circuit interconnect structures is their inability to be adaptable to and dynamically optimized for a range of specific tasks. Another problem with conventional integrated circuit interconnect structures is their inability to be dynamically *changed* to precisely meet the requirements of the current task or set of tasks at hand. While the programmable interconnect within FPGA-type devices can in theory be reconfigured, in practice this is rarely done because of the complexity of the tools required and the time lapse ("configuration latency") associated with such changes. Another problem with conventional integrated circuit interconnect structures is that they are not well-suited for a semiconductor architecture wherein the data and controls paths are constantly changing. Such changes are desirable in order to support real-time optimization of the circuitry and interconnect to match the needs of the task or set of tasks at hand.

While the interconnect structures of the prior art may be suitable for the particular purpose to which they address, they are not as suitable for providing a flexible, efficient, self-routing and dynamically optimized means for connecting together functional elements within a semiconductor device.

In these respects, the self-routing, message-based interconnect system for electrical devices according to the present invention substantially departs from the conventional concepts and designs of the prior art, and in so doing provides an apparatus primarily developed for the purpose of providing a flexible, efficient, self-routing and dynamically optimized means for connecting together functional elements within a semiconductor device.

SUMMARY OF THE INVENTION

In view of the foregoing disadvantages inherent in the known types of conventional integrated circuit interconnect structures now present in the prior art, the present invention provides a new self-routing, message-based interconnect system for electrical construction wherein the same can be utilized to provide a flexible, efficient, self-routing and dynamically optimized means for connecting together functional elements within a semiconductor device.

The general purpose of the present invention, which will be described subsequently in greater detail, is to provide a new self-routing, message-based interconnect system for electrical devices that has many of the advantages of the conventional integrated circuit interconnect structures mentioned heretofore and many novel features that result in a new self-routing, message-based interconnect system for electrical devices which is not anticipated, rendered obvious, suggested, or even implied by any of the prior art integrated circuit interconnect types, either alone or in any combination thereof.

To attain this, the present invention generally comprises a series of processing and/or logic elements to be interconnected; a series of primary and secondary groups (busses) of interconnect paths including local and/or long distance busses with separate signal paths for data, address, and/or control signals; and arbitration and control circuitry including a randomization element.

The PROCESSING BLOCK is one of a series or array of elements to be interconnected by the present invention. It can consist of one or more of the following components or any combination thereof: Central Processing Units (CPU), Arithmetic Logic Units (ALU); Memory Elements (MEM); Arbitrary Function Generators (ARB); State Machines; Digital Signal Processors (DSP); Programmable Logic Devices (PLD including Field Programmable Gate Array (FPGA) and Complex PLD (CPLD)); and/or General Purpose logic. The array of PROCESSING BLOCKS can either be homogeneous or non-homogeneous. The actual type of blocks being interconnected is not what is being considered in the scope of this invention; it is the method for dynamically connecting the blocks together.

The INTERCONNECT BUSSES provide the physical connections over which data -- including applications data, addressing, control and signaling information -- is passed between the PROCESSING BLOCKS. In the preferred embodiment, the INTERRCONNECT BUSSES are broken into two, independent types; the LOCAL busses and the LONG DISTANCE busses. Each PROCESSING BLOCK has both input and output LONG DISTANCE busses for each of the four directions (a total of 8 LONG DISTANCE busses for this example in two-dimensional space, although higher-order multi-dimensional implementations are to be considered within the scope of this patent.) The preferred embodiment also employs but is not limited to twelve dedicated bus structures for LOCAL (nearest neighbor) connections. These LOCAL busses operate

independently from the LONG DISTANCE bus structures and serve primarily to offer dedicated, high-bandwidth communications between neighboring PROCESSING BLOCKS which minimizes the traffic that must traverse the LONG DISTANCE networks.

The ARBITRATION and CONTROL CIRCUITRY performs three distinct functions. First, it is responsible for formatting message requests that originate in a given PROCESSING BLOCK and forwarding these messages accordingly; second, it detects incoming messages from other blocks and determines the availability of a path that would move the incoming message closer to its destination and -- if more than one such path exists -- selects one of the available paths and forwards the message down that path, adjusting the addressing information accordingly. If a path is NOT available, this circuitry prevents the incoming message from selecting this path, thereby forcing a different path to be established. Third, the ARBITRATION and CONTROL CIRCUITRY is also responsible for determining that an incoming message has reached its destination; if the message is at its destination, the ARBITRATION and CONTROL CIRCUITRY first checks that the resource being requested in the message header is available, and, if so, returns an ACKNOWLEDGE signal. If no acknowledgement is received by the source within a set period of time, the host will issue re-tries as described later in this document. After the path is established, the incoming message payload (data) is placed into the desired resource(s) by the ARBITRATION and CONTROL CIRCUITRY.

Each time a message attempts to establish a path through a given messenger, a state bit within the messenger's RANDOMIZATION BLOCK toggles to the opposite state. The state bit is toggled regardless of whether the attempt to establish the path was successful or not. The state bit controls which of the two output ports (at most, two output ports will move a given message closer to its destination) will be "tried" first when attempting to establish a message path. This insures that subsequent retries (should they be required) will automatically attempt to employ different paths/resources on each retry. The RANDOMIZATION BLOCK also includes a small pseudo-random number generator implemented as a maximal length linear-feedback shift register (LFSR). The starting code and pattern for this LFSR is a function of the physical row/column information associated with the individual PROCESSING BLOCK with which it is associated. When a message path cannot be established or if the message destination is busy, the message source will retry the communication repeatedly, first by delaying 1, 2, 3, and then 4 clock cycles between reties. Should the communication channel still not be available after the initial try and the four, sequentially increasing clock-cycle based delays as described, the random count of the LFSR is then used to determine the number of clock cycles to delay before subsequent retries. This randomization provides a mechanism that eliminates the opportunity for deadlock situations, wherein multiple sources repeatedly attempt to communicate with a single destination or otherwise repeatedly compete for a resource. This randomization – after the initial, aggressive 1,2,3, and 4 clock cycle retry sequence – assures that no two subsequent retry sequences and timing will be identical. This provides intrinsic load-leveling and maximizes channel and resource availability. It should be obvious to anyone skilled in the art that the RANDOMIZATION BLOCK can

be configured and implemented in many other ways than that described in this disclosure. As such, the present invention is not limited to the selected method described herein.

There has thus been outlined, rather broadly, the more important features of the invention in order that the detailed description thereof may be better understood, and in order that the present contribution to the art may be better appreciated. There are additional features of the invention that will be described hereinafter.

In this respect, before explaining at least one embodiment of the invention in detail, it is to be understood that the invention is not limited in its application to the details of construction and to the arrangements of the components set forth in the following description or illustrated in the drawings. The invention is capable of other embodiments and of being practiced and carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein are for the purpose of description and should not be regarded as limiting.

A primary object of the present invention is to provide a self-routing, message-based interconnect system for electrical devices that will overcome the shortcomings of the prior art devices.

Another object of the present invention is to provide a self-routing, message-based interconnect system for electrical devices to provide a flexible, efficient, and dynamically optimized means for connecting together functional elements within a semiconductor device.

Another object is to provide a self-routing, message-based interconnect system for electrical devices that allows the data and control paths within an integrated circuit to be quickly and dynamically modified -- in real time -- to precisely match the requirements of the current task or set of tasks that the device is to perform.

Another object is to provide a self-routing, message-based interconnect system for electrical devices that minimizes contention for resources.

Another object is to provide a self-routing, message-based interconnect system for electrical devices that allows multiple connection paths for multiple, concurrent tasks to be quickly and automatically discovered and selected.

Another object is to provide a self-routing, message-based interconnect system for electrical devices that dynamically varies the path-selection decision tree in order to prevent deadlock.

Another object is to provide a self-routing, message-based interconnect system for electrical devices that automatically releases resources after a transfer is complete to allow these resources to be used by other tasks if needed.

Another object is to provide a self-routing, message-based interconnect system for electrical devices that automatically selects paths with minimal physical lengths and propagation delays.

Another object is to provide a self-routing, message-based interconnect system for electrical devices that automatically senses blocked paths, and -- if so blocked -- first attempts to select an alternate, equally beneficial path or, in the event no such optimal path exists, automatically performs a "re-try" with different selection criteria asserted.

Other objects and advantages of the present invention will become obvious to the reader and it is intended that these objects and advantages are within the scope of the present invention.

To the accomplishment of the above and related objects, this invention may be embodied in the form illustrated in the accompanying drawings, attention being called to the fact, however, that the drawings are illustrative only, and that changes may be made in the specific construction illustrated.

BRIEF DESCRIPTION OF THE DRAWINGS

Various other objects, features and attendant advantages of the present invention will become fully appreciated as the same becomes better understood when considered in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the several views, and wherein:

Figure 1 is a Block Diagram of an Array of Processing Blocks with an Expanded View of the individual connections to/from the Messenger of one such block.

Figure 2 illustrates the functions of the bit-fields in a message address and provides a specific example (for a message that is to travel three columns to the left and five rows down). The example provides three bits for both Row and Column offset, along with a sign bit for indicating directionality (up versus down, left versus right). This provides for an addressable range of movement of 2^3-1 rows and columns. (the "minus 1" accounts for an offset of zero, meaning the message is in the destination row and/or column). Note that for a larger array of Processing Blocks, the bit fields required for addressing would be extended or contracted as required.

Figure 3 illustrates four possible paths from Processing Block 1,1 (e.g. "Row 1, Column 1") to Processing Block 2,4 (the first number is the ROW, second number is the COLUMN).

Figure 4 illustrates the iterative process for finding/establishing a Long Distance Message Channel.

Figure 6 illustrates an automatic retry.

Figure 7 is a flow chart for the Decision Tree for incoming messages (such logic resides in every Messenger Block in every Processing Block in the system).

Figure 7 is representative 4-Bit Maximal Length Linear Feedback Shift Register (LFSR) for pseudorandom number generation (part of the Arbitration and Randomization logic in each Messenger block).

Figure 8 demonstrates how it is possible for a total of sixteen 32-bit Messages to be passed into, through, and/or out of a single Processing Block simultaneously.

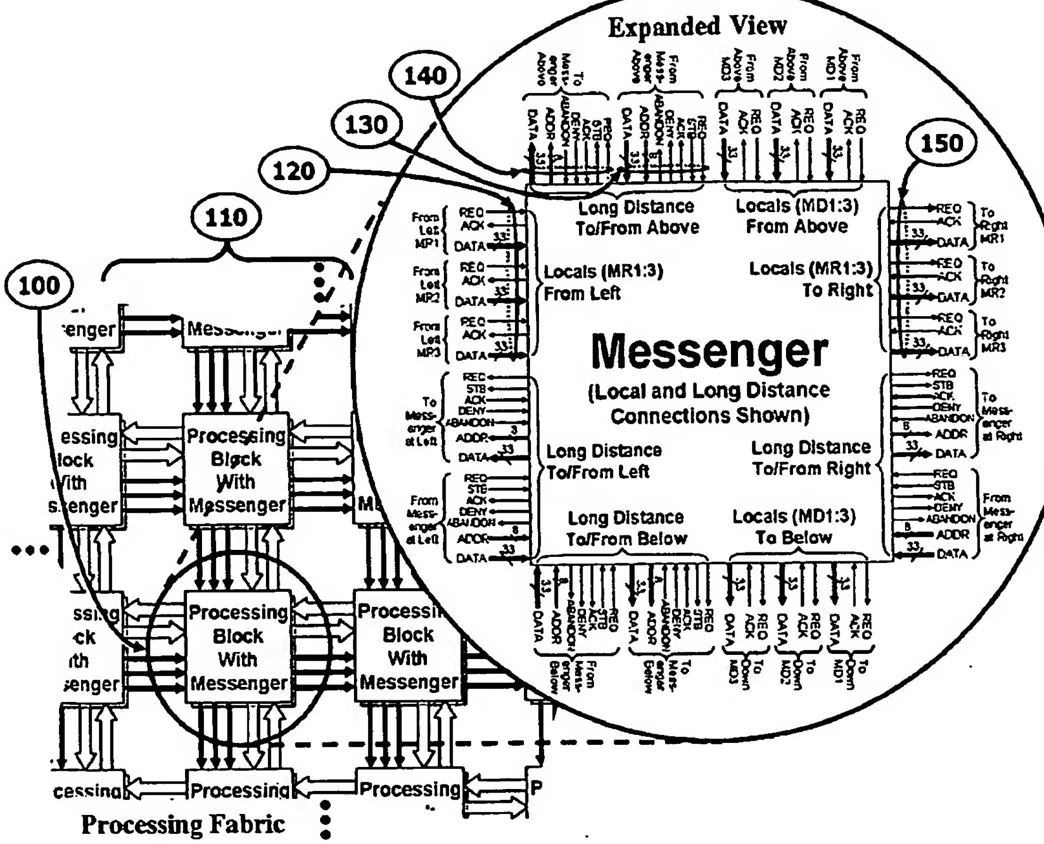


Figure 1. Block Diagram of an Array of Processing Blocks with an Expanded View of the Individual Connection To/From One Block

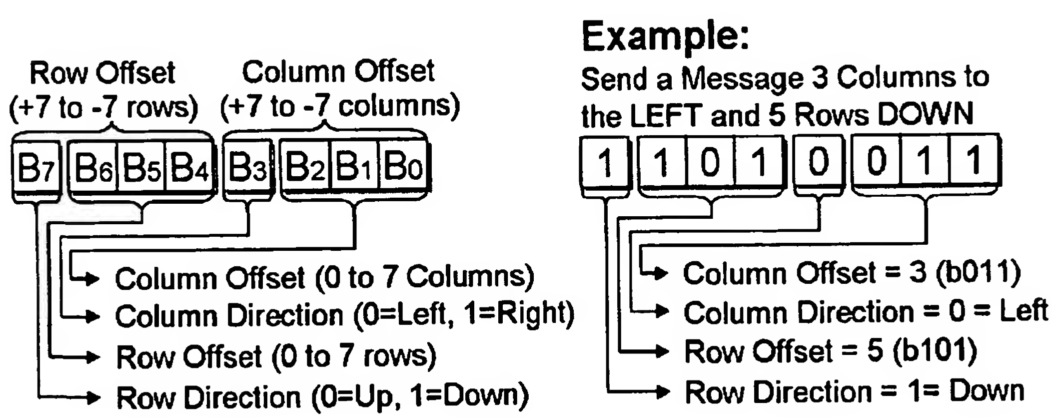


Figure 2. Functions of the bit-fields in a Message Address

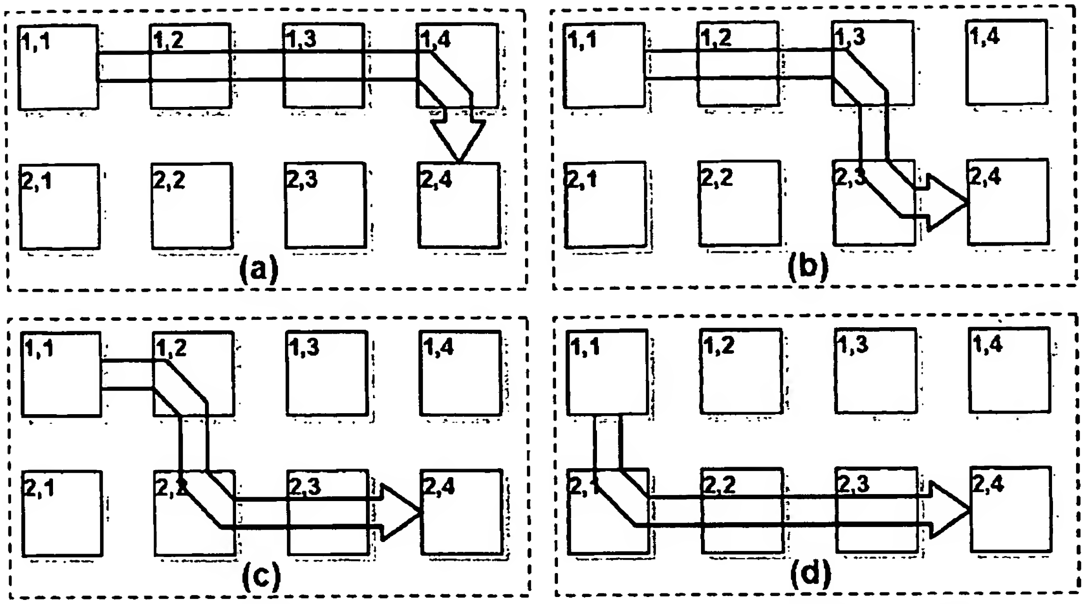
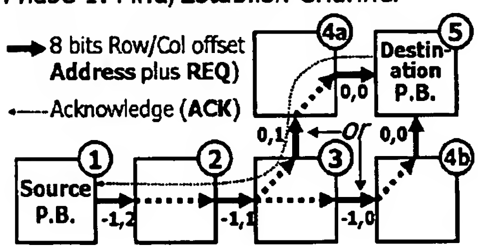


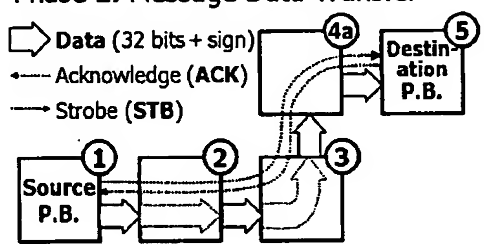
Figure 3. Four Possible Paths from Processing Block 1,1 to Processing Block 2,4
(first number is ROW, second number is COLUMN)

Phase 1. Find/Establish Channel



1. The Messenger in the Source Processing Block ① requests a transmission channel in the horizontal direction by asserting a message with address -1,2 and driving the associated REQ line
2. Row/Col Offset Address propagates through ②, Col offset decrements
3. At ③, Col = Row. If one direction is occupied, the other automatically "wins", else the direction is determined by state of toggle bit (randomized)
4. Request passed through 4a or 4b to destination Processing Block; if the destination is ready / available, ACK is returned along the same path

Phase 2. Message Data Transfer



1. Once a channel is established (ACK from the destination ⑤ received at the source ①), Sources places first data on output port/bus and asserts STB
2. This channel is then dedicated to this transaction for the duration of the transfer (Direct and other, orthogonal transfers can take place through the same Processing Blocks simultaneously)
3. The source ⑤ generates STB to clock data into the destination ⑤; this continues until the transfer is complete
4. Finally, the source ① drops REQ; this releases all involved resources to be used as needed for additional transfers

Figure 4. Process for Finding/Establishing a Long Distance Message Channel

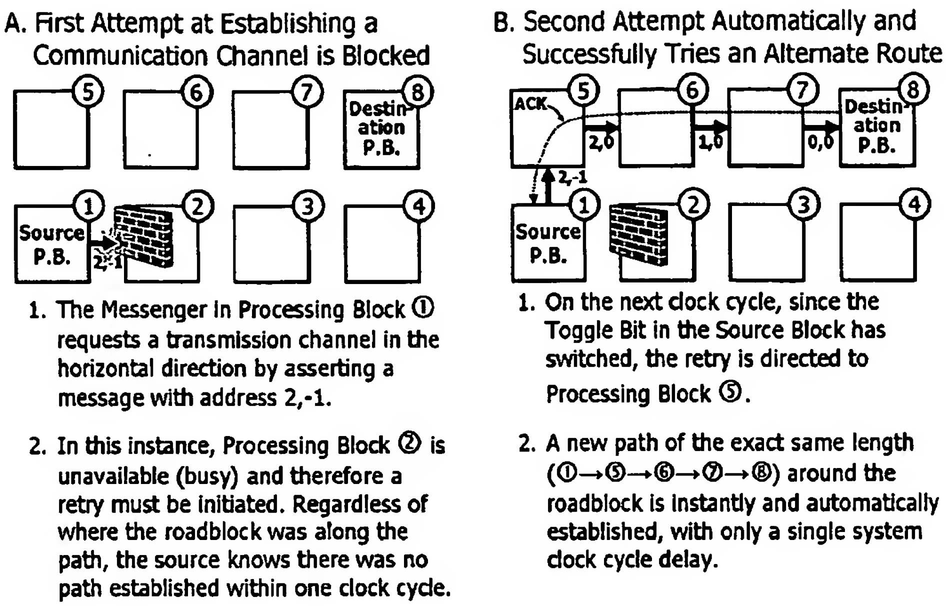


Figure 5. Automated Retry Illustrated

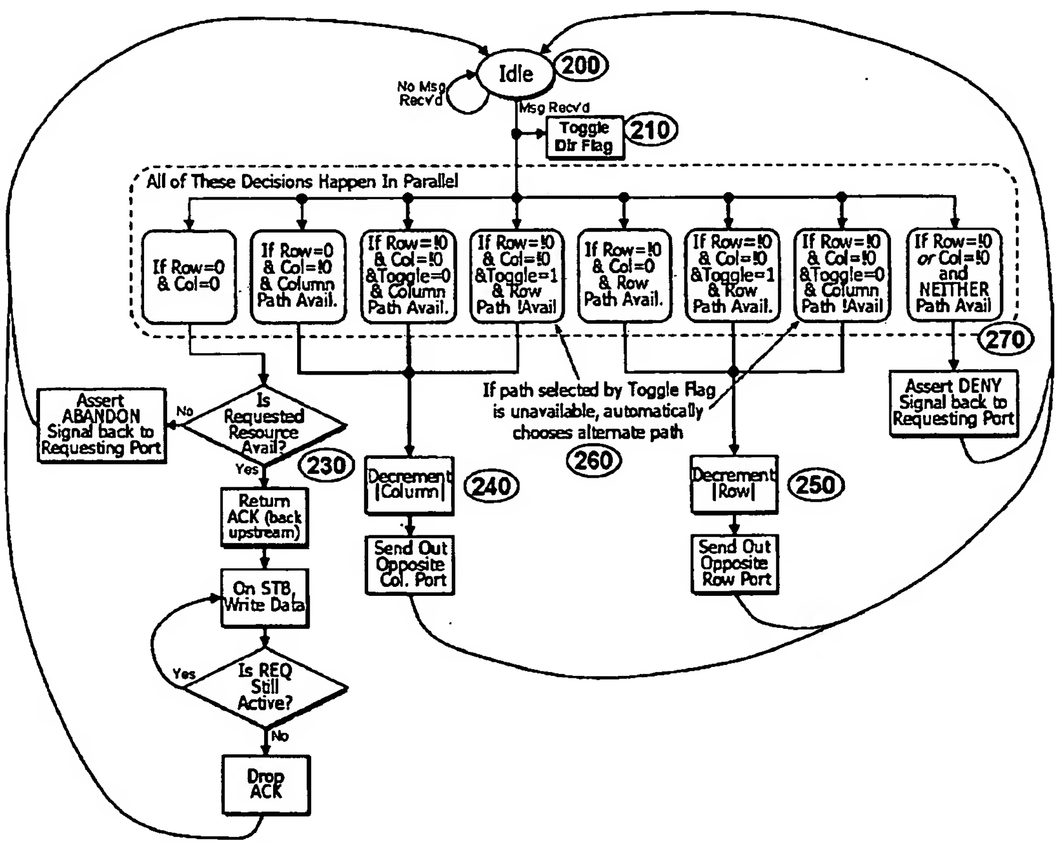
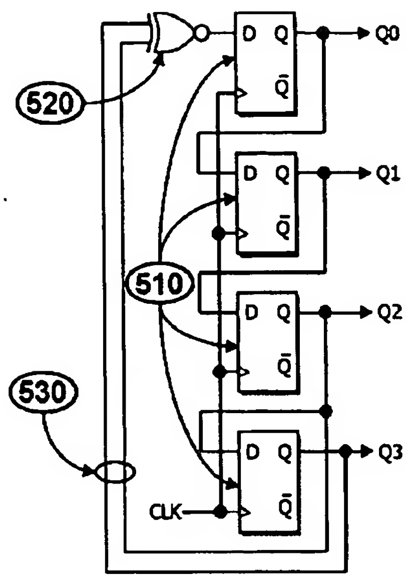


Figure 6. Decision Tree for Incoming Messages (Such logic resides in every Messenger Block in every Processing Block in the system)



Signal	Q 0	Q 1	Q 2	Q 3	XNor	Hex
Clk 0	0	0	0	0	1	0
Clk 1	1	0	0	0	1	1
Clk 2	1	1	0	0	1	3
Clk 3	1	1	1	0	0	7
Clk 4	0	1	1	1	1	E
Clk 5	1	0	1	1	1	D
Clk 6	1	1	0	1	0	B
Clk 7	0	1	1	0	0	6
Clk 8	0	0	1	1	1	C
Clk 9	1	0	0	1	0	9
Clk 10	0	1	0	0	1	2
Clk 11	1	0	1	0	0	5
Clk 12	0	1	0	1	0	A
Clk 13	0	0	1	0	0	4
Clk 14	0	0	0	1	0	8

Figure 7. 4-Bit Maximal Length Linear Feedback Shift Register (LFSR) for Pseudorandom Number Generation (Part of the Arbitration and Randomization logic)

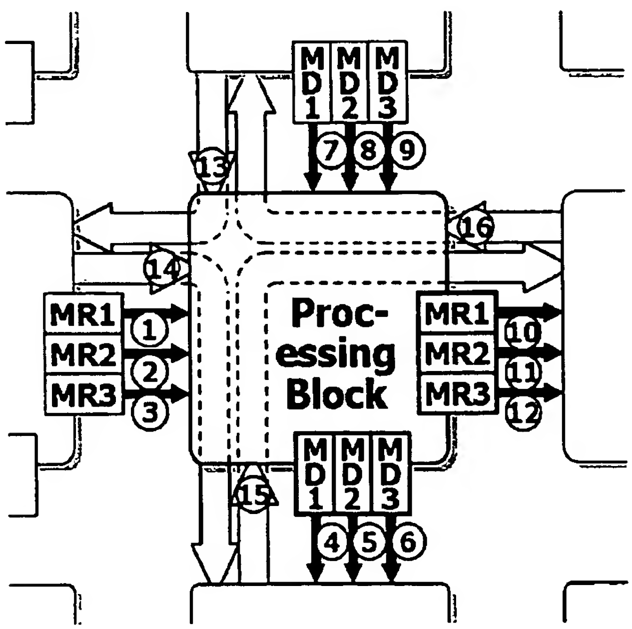


Figure 8. Sixteen Possible, Simultaneous Messages Being Passed Into, Through, and/or Out of a Single Processing Block

DESCRIPTION OF THE PREFERRED EMBODIMENT

Turning now descriptively to the drawings, in which similar reference characters denote similar elements throughout the several views, the attached figures illustrate a self-routing, message-based interconnect system for electrical devices, which is comprised of a series (110) of processing and/or logic blocks (100) to be interconnected; a series of primary and secondary groups (busses) of interconnect paths including local and/or long distance busses with separate signal paths for data, address, and/or control signals (120, 130, 140, and 150); arbitration and control circuitry including a randomization element.

The PROCESSING BLOCK (100) is one of a series or array (110) of elements to be interconnected by the present invention. It can consist of one or more of the following components or any combination thereof: Central Processing Units (CPUs), Arithmetic Logic Units (ALU); Memory Elements (MEM); Arbitrary Function Generators (ARB); State Machines; Digital Signal Processors (DSP); Programmable Logic Devices (PLD including Field Programmable Gate Array (FPGA) and Complex PLD (CPLD)); and/or General Purpose logic. The actual type and number of blocks being interconnected is not central to or considered in the scope of this invention; it is the method for dynamically connecting the blocks together.

The INTERCONNECT BUSSES (120, 130, 140, 150) provide the physical connections over which data -- including applications data, addressing, and control information -- is passed between the PROCESSING BLOCKS. In the preferred embodiment; the INTERRCONNECT BUSSES are broken into two, independent types; the LOCAL busses (120, 150) and the LONG DISTANCE busses (130, 140). Each PROCESSING BLOCK has both input (130) and output LONG DISTANCE (140) busses for each of the four directions (a total of 8 LONG DISTANCE busses for this example in two-dimensional space, although higher-order multi-dimensional implementations are covered by this patent.) The preferred embodiment also employs twelve dedicated bus structures (120, 150) for LOCAL (nearest neighbor) connections. These LOCAL busses operate independently from the LONG DISTANCE bus structures and serve primarily to minimize the traffic that must traverse the LONG DISTANCE networks. The LOCAL busses in the preferred embodiment are uni-directional, with separate ports, control, and interconnect for inputs (120) and outputs (150), but this is not to be construed as limiting.

The ARBITRATION and CONTROL CIRCUITRY (Figure 6 shows the Decision Tree Diagram; Figure 7 illustrates a representative 4-bit Linear Feedback Shift Register that is used to introduce randomization for the retry sequence) performs three distinct functions: First, it is responsible for formatting message requests that originate in a given PROCESSING BLOCK and forwarding these messages accordingly; second, it detects incoming messages from other blocks and determines the availability of a path that would move the incoming message closer to the destination and -- if more than one such path exists -- selects one of the available paths and forwards the message down that path, adjusting the addressing information accordingly. If a path is NOT available, this

circuitry prevents the incoming message from selecting this path, forcing a different path to be established. Third, the Arbitration and Control block is also responsible for determining that an incoming message has reached its destination (230); if the message is at its destination, the Arbitration and Control circuitry first checks that the resource being requested in the message header is available, and, if so, returns an ACKNOWLEDGE signal. If no acknowledgement is received by the source within a set period of time, the host will issue re-tries as to be described later in this document. After the path is established, the incoming message payload (data) is placed into the desired resource(s) by the Arbitration and Control circuitry. The signal names and polarities used in describing the ARBITRATION and CONTROL CIRCUITRY are not an object of the current invention. Similarly, such arbitration and control could be realized through a different set of signals or control parameters. Another possible functional variation of this invention involves buffering messages using data registers, memory locations, and/or first-in/first-out i.e. FIFO memory.

Each time a message attempts to establish a path through a given messenger, a state bit (210) within the messenger's RANDOMIZATION BLOCK toggles to the opposite state. The state bit is toggled regardless of whether the attempt to establish the path was successful or not. At most, in a two-dimensional routing plane as described in the preferred embodiment, only two of the potential output ports would move a given message closer to its destination. The state bit controls which of these two output ports will be "tried" when attempting to establish a message path. This insures that subsequent retries (should they be required) will automatically attempt to employ *different* paths/resources on each retry. The RANDOMIZATION BLOCK also includes a small pseudo-random number generator implemented as a maximal length linear-feedback shift register (LFSR, see Figure 7). The starting code for this LFSR is a function of the physical row/column information for with the individual PROCESSING BLOCK with which it is associated. When a message path cannot be established or if the message destination is busy, the message source will retry the communication repeatedly, first by delaying 1, 2, 3, and then 4 clock cycles between reties. Should the communication channel still not be available after the initial try and the four, sequentially increasing clock-cycle based delays as described, the random count of the LFSR is then used to determine the number of clock cycles to delay before subsequent retries. This randomization provides a mechanism that eliminates the opportunity for deadlock situations, wherein multiple sources repeatedly attempt to communicate with a single destination or otherwise repeatedly compete for a resource. The randomization -- after the initial, aggressive 1,2,3, and 4 clock cycle retry sequence -- assures that no two subsequent retry sequences and timing will be identical. This provides intrinsic load-leveling and maximizes channel and resource availability. This provides intrinsic load-leveling and maximizes channel and resource availability. The randomization block consists of two primary sub-blocks; the toggle state bit which picks a path when two possible paths exist and the Linear Feedback Shift Register (LFSR) which provides a pseudo-random number for varying the number of machine cycles that pass before retry attempt is initiated in the case a connection cannot be established after an initial, predetermined sequence of attempts. In the case an attempt to establish a message path fails, the retry logic in the current invention first performs an aggressive series of retries

waiting first one, then 2, then 3, and finally 4 clock cycles between successive retries. After these five attempts (the initial attempt and then four, sequentially increasingly delayed attempts), the current value of the pseudo-random LFSR is used to determine the number of clock cycles to delay before making another retry. The retry sequence of 1,2,3 and then 4 clock cycles and then -- if a connection is still not established -- moving to the pseudorandom sequence is simply a protocol developed for a specific set of applications and the anticipated message traffic those applications would generate. This pattern is not an object of this invention and other potentially useful sequences could be employed. Alternately, a free-running (not synchronized to any system clock or event) counter or shift register could be employed.

The PROCESSING BLOCK is one of a series or array of elements to be interconnected by the present invention. It can consist of one or more of the following components or any combination thereof: Arithmetic Logic Units (ALU); Memory Elements (MEM); Arbitrary Function Generators (ARB); State Machines; Digital Signal Processors (DSP); Programmable Logic Devices (PLD). The actual type of blocks being interconnected is not what is being considered in the scope of this invention; it is the method for dynamically connecting the blocks together. The interconnect busses (for both local and long distance messaging as described below) connect to the PROCESSING BLOCKS and provide them with digital information upon which to act. The preferred embodiment of the current invention employs an essentially square, two-dimensional PROCESSING BLOCK. The shape of the PROCESSING BLOCK is not a fundamental attribute of this invention and should in no way be construed as limiting. Rectangular, triangular, round, hexagonal, other polygon-shaped and even irregularly shaped PROCESSING BLOCKS of two, three and even higher-order multi-dimensional nature and construction should be considered within the scope of this invention.

The INTERCONNECT BUSSES provide the physical connections over which data -- including applications data, addressing, and control information -- is passed between the PROCESSING BLOCKS. In the preferred embodiment; the INTERRCONNECT BUSSES are broken into two, independent types; the LOCAL busses and the LONG DISTANCE busses. Each PROCESSING BLOCK has both input and output LONG DISTANCE busses for each of the four directions (a total of 8 LONG DISTANCE busses for this example in two-dimensional space, although three and even higher-order multi-dimensional implementations are covered by this patent.)

The preferred embodiment also employs twelve dedicated bus structures for LOCAL (nearest neighbor) connections. These LOCAL busses operate independently from the LONG DISTANCE bus structures and serve primarily to minimize the traffic that must traverse the LONG DISTANCE networks. The interconnect busses for both LOCAL and LONG DISTANCE messaging (as described below) connect the PROCESSING BLOCKS together and to other resources (external Input/Output ports, memory blocks, etc.) and provide them with digital information upon which to act. The Interconnect Busses also provide paths for data that is calculated or otherwise produced from a source PROCESSING BLOCK (or an external data source) to destinations -- either other PROCESSING BLOCKS within the device or to destinations external to the

device. In the current embodiment, LOCAL messages are passed via twelve unidirectional 32-bit wide datapaths connect nearest-neighbor PROCESSING BLOCKS (refer to the definition and possible permutations of nearest neighbor connections, below). Of the twelve, six such busses are inputs to the PROCESSING BLOCK and six are outputs. For the purpose of this description, assume an essentially square, two-dimensional PROCESSING BLOCK wherein 3 of the 32-bit input busses noted enter the PROCESSING BLOCK at the "top" from the PROCESSING BLOCK above; three enter at the "left" from the PROCESSING BLOCK to the left. Similarly, the LOCAL outputs from the PROCESSING BLOCK are three 32-bit busses exiting on the "right" and three on the "bottom". The terms "left, right, top, and bottom" are arbitrary and meant only to suggest the orthogonal relationship between the ports. The LOCAL bus connections are simple, unidirectional ports. A "Presence" bit in the associated destination register of the destination PROCESSING BLOCK provides an indication to the source PROCESSING BLOCK that the destination is available to accept new data. The source PROCESSING BLOCK checks the Presence bit before writing its information into the destination register. This hardware handshaking and the exclusivity of these LOCAL interconnect resources creates three highly-available paths for nearest-neighbor communication. Studies have clearly shown that the majority of data communication in multi-processing environments is local in nature. The number and dedicated nature of the LOCAL interconnect busses in the current invention reflect that concentration of local communications and free the LONG DISTANCE messenger system for more global (and therefore possibly contentious) communications.

In the current invention, the LONG DISTANCE Interconnect busses are also 32-bits wide (data path) and unidirectional. In addition to the 32 data bits, 10 bits of address information, a Request (REQ) signal, an ACKNOWLEDGE (ACK) signal, a STROBE (STB) signal and a polarity (sign) bit are also employed and are physically grouped with the 32 data signal paths with which they are associated. Each side of each PROCESSING BLOCK has two sets of such busses -- one for input and one for output -- connected to the LONG DISTANCE Messenger circuitry of the logical "nearest neighbor" blocks. For the purposes of this disclosure, the term "nearest neighbor" is defined from a logical and not necessarily physical perspective. Loops, toruses, hyper-toruses, hyper-cubes and other geometric permutations -- wherein physical nearest neighbor PROCESSING BLOCKS are not necessarily nearest logical neighbors -- should also be considered within the scope of this disclosure. Additionally, the terms "left, right, top, and bottom" are arbitrary and meant only to suggest the orthogonal relationship between the ports/busses. The terms "North, South, East, and West" or other permutations are equally acceptable. As noted in the description of the PROCESSING BLOCKS, the shape of the PROCESSING BLOCK and the number of ports attached to said blocks are not specific objects of this disclosure. Likewise, the number of ports/busses, the number and types of signals and interconnects grouped to form the busses, nor the polarity nor nomenclature of the signals involved should be seen as limiting. Specifically, the invention disclosure includes single-bit (i.e. "serial") connections and ports as well as multi-bit busses and ports both narrower and wider in bit-width than the 32-bit busses described in the preferred embodiment. Non-binary (non-

digital), multi-state, analog, optical, and/or any other time-variant signaling mechanisms capable of carrying information are also within the scope of this invention.

The messenger system in the current invention discovers and creates high bandwidth, reconfigurable and self load-leveling data paths on an as-needed basis. This provides dynamically optimized communications channels between the PROCESSING BLOCKS and other components in a system. The functionality and number of PROCESSING BLOCKS to communicate via the present invention are outside of the scope of this disclosure; it is the means and apparatus used for dynamically discovering and creating/relinquishing these communications channels that is the focus of this disclosure.

In the preferred embodiment, separate, distinct communications channels exist for local communications and for long-distance (two or more PROCESSING BLOCKS away) communications. The LOCAL CONNECTIONS provide the bulk of PROCESSING BLOCK to PROCESSING BLOCK communication. They include six unidirectional input busses, three entering the "top" of the PROCESSING BLOCK and three entering the "left" side of the block. Similarly, each PROCESSING BLOCK drives six output busses; three to the "right" and three "down". As previously noted, the number, type, construction and directionality (unidirectional or bidirectional) are not an object of this disclosure. In the preferred embodiment, these LOCAL Message busses consist of 33 data signals (a 32-bit data bus plus a sign bit for indicating polarity of the 32-bit data word e.g. "positive" or "negative") plus two signal lines for simple hardware handshaking. These two signals REQ (for "Request") and "ACK" (for "Acknowledge") operate as follows: When a source PROCESSING BLOCK has data that it wishes to pass to it's neighboring block (either to the right or down), the source PROCESSING BLOCK places its data (including sign bit) on the 33-bit wide data bus and then asserts the associated REQ signal. If the destination PROCESSING BLOCK is able to accept the data, it does so immediately and responds with an ACK signal which it sends s back to the source PROCESSING BLOCK. If for any reason the destination PROCESSING BLOCK can NOT accept the data at that time, it simply does not return the ACK signal which, in turn, forces the source PROCESSING BLOCK to initiate one or a series of "retries" as described below. All six potential Local outputs can be driven simultaneously and all six local input channels can also accept data, simultaneously, in the same system cycle.

LONG DISTANCE connection busses are similar in structure in that they employ 33 data signals (32 data plus a sign bit), a Request (REQ) and an Acknowledge (ACK) signal. However, the Long Distance busses also contain: a Strobe (STB) signal for clocking information into the destination once a communication channel is established; a DENY signal for refusing a connection request; an ABANDON signal that the destination PROCESSING BLOCK can assert in the event it is busy and unable to service the request; and with 8 Address (ADDR7:0) lines that indicate the row and column offset of the requested destinaio0n from the source. Instead of enforcing a "top-to-bottom" and "left-to-right" information flow as dictated by the directionality of the

LOCAL connection busses, the Long Distance connection busses provide separate, distinct input and output busses for each of the four generalized directions (e.g. up, down, left, and right). As previously noted, the nomenclature, polarity, and number of the actual, physical signal lines described herein is simply a preferred embodiment of the invention; other realizations of the functionality of this invention are possible and are considered within the scope of this disclosure.

When a PROCESSING BLOCK wishes to initiate a Long Distance message, it first places the address -- expressed as a relative row/column offset from the source to its requested destination -- on its address outputs and drives its REQ line active. Information regarding the type, length, and other characteristics of the ensuing message are also encoded into the first data word which is also presented on the data lines at this time. If the destination is in the same logical row or column as the source, the output port in that same row or column and closets to the destination is used. If the destination does NOT share the same logical row or column as the source, the source can choose either of the two ports that are in the direction of the destination. Which of the two ports it chooses is determined by the state of the TOGGLE BIT resident in the source messenger; the toggle bit changes state every time a message is initiated (either an initial try or any subsequent retry of a given message) by a PROCESSING BLOCK and every time a message -- either successful or unsuccessful attempts -- is passed through the messenger associated with that PROCESSING BLOCK. This provides a degree of randomness to the selection of paths through the array and also insures that different paths will be "tried" on subsequent retries, increasing the likelihood of discovering an available route in the event a retry is required.

Once the data, address, and REQ signals for a new message are driven out of the source PROCESSING BLOCK, the next messenger along the selected path (determined by the toggle bit in the source as described above) accepts those inputs and responds based on the availability of resources. Four possible scenarios exist:

1. The Row and Column address offsets are both ZERO (destination has been reached).
2. The Row offset is ZERO, the Column offset is NON-zero (the message has reached the ultimate destination ROW, but still needs to move to the correct COLUMN).
3. The Column offset is ZERO, the Row offset is NON-zero (the message has reached the ultimate destination COLUMN, but still need to move to the correct ROW).
4. Neither the Row *nor* Column Offsets are ZERO (the message needs to continue in both the row and column directions to reach the destination).

In the first scenario (row AND column offsets are both ZERO), the PROCESSING BLOCK recognizes from the zero offset address that it is the destination. It then uses the information in the data part of the message to determine what resource(s) is/are being requested and checks if the requested resource(s) is/are available. If the requested resource is available, the destination PROCESSING BLOCK responds with an ACK signal back through the same port from whence the REQ originated. Note that it is unlikely that near-neighbor PROCESSING BLOCKS – with a source and destination

logically adjacent – would use the LONG DISTANCE resources. Such communications would more likely use the direct, dedicated LOCAL communications paths.

In the second and third scenarios (row OR column offset is zero, but not BOTH), the current PROCESSING BLOCK's messenger attempts to forward the message along in the non-zero dimension, decrementing the associated row or column offset address seen on the Address lines as the message passes through. In addition and as previously mentioned, the Toggle Bit in the current messenger block is toggled (switched to the opposite state.) This changes the "preferred" direction for message forwarding should the next message passing through this block have a choice of paths.

In the fourth scenario (neither the row NOR column offset is zero), the current PROCESSING BLOCK's messenger attempts to forward the message along in the one of non-zero dimensions, decrementing the associated row or column offset address seen on the Address lines as the message passes through. Note the Toggle Bit as described earlier is used to determine which of two potentials paths should be employed. In the event a sourcing PROCESSING BLOCK should NOT receive an ACK signal back within a prescribed time period (in the case of the preferred embodiment, within a single system clock period), a retry of the same message will be initiated. Note that this no-ACK condition could result from either channel congestion (no available channels currently exist to enable the message to be successfully forwarded from the source to the destination) or the destination itself may be busy. Regardless of the reason for the retry, the Toggle Bit in each Messenger in each PROCESSING BLOCK in a potential message path insures that -- at EVERY routing decision point between the source and the destination -- a different path will be attempted in any/all subsequent attempts to establish a path through that PROCESSING BLOCK.

In the preferred embodiment of this invention, a source PROCESSING BLOCK, when faced with the need to attempt a retry of a given message, will first immediately retry the message, then, if still not successful, will wait one system clock cycle before making a third attempt. If required, a fourth attempt is delayed two clock cycles, a fifth, three clock cycles, etc. until four retry attempts have been issued and were unsuccessful. Beginning with the ninth retry, the clock delay between retry attempts is based on the value (0 to 15 clock cycles) of a pseudo-random number generator (implemented in the preferred embodiment as a four bit linear-feedback-shift register or LFSR). This LFSR -- like the Toggle Bit, is advanced each time a message -- successful or not -- is initiated or passed through the PROCESSING BLOCK within which this messenger resides. The introduction of this randomization after the initial, aggressive retry pattern (1,2,3, etc clock cycles) prevents two or more processes from simultaneously and in lock-step issuing conflicting message attempts. Preventing this phenomenon -- referred to as "deadlock" -- is a significant feature of the current invention. Note however that the aggressive-then-random retry pattern is not unique; other retry patterns and randomization techniques could be employed and all are covered within the scope of this disclosure.

As to a further discussion of the manner of usage and operation of the present invention, the same should be apparent from the above description. Accordingly, no further discussion relating to the manner of usage and operation will be provided.

With respect to the above description then, it is to be realized that the optimum dimensional relationships for the parts of the invention, to include variations in size, materials, shape, form, function and manner of operation, assembly and use, are deemed readily apparent and obvious to one skilled in the art, and all equivalent relationships to those illustrated in the drawings and described in the specification are intended to be encompassed by the present invention.

Therefore, the foregoing is considered as illustrative only of the principles of the invention. Further, since numerous modifications and changes will readily occur to those skilled in the art, it is not desired to limit the invention to the exact construction and operation shown and described, and accordingly, all suitable modifications and equivalents may be resorted to, falling within the scope of the invention.

ABSTRACT OF THE DISCLOSURE

A self-routing, message-based interconnect system for electrical devices that provides a flexible, efficient, and dynamically optimized means for connecting together functional elements within a semiconductor device. The inventive device includes a series of processing and/or logic elements to be interconnected; a series of primary and secondary groups (busses) of interconnect paths including local and/or long distance busses with separate signal paths for data, address, and/or control signals; arbitration and control circuitry including a randomization element.

The PROCESSING BLOCK is one of a series or array of elements to be interconnected by the present invention. It can consist of one or more of the following components or any combination thereof: Central Processing Units (CPU); Arithmetic Logic Units (ALU); Memory Elements (MEM); Arbitrary Function Generators (ARB); State Machines; Digital Signal Processors (DSP); Programmable Logic Devices (PLD). The actual type of blocks being interconnected is not what is being considered in the scope of this invention; it is the method for dynamically connecting the blocks together.

The INTERCONNECT BUSSES provide the physical connections over which data -- including applications data, addressing, and control information -- is passed between the PROCESSING BLOCKS. In the preferred embodiment; the INTERRCONNECT BUSSES are broken into two, independent types; the LOCAL busses and the LONG DISTANCE busses. Each PROCESSING BLOCK has both input and output LONG DISTANCE busses for each of the four directions (a total of 8 LONG DISTANCE busses for this example in two-dimensional space, although three and even higher-order, multi-dimensional implementations are covered by this patent.) The preferred embodiment also employs twelve dedicated bus structures for LOCAL (nearest neighbor) connections. These LOCAL busses operate independently from the LONG DISTANCE bus structures and serve primarily to minimize the traffic that must traverse the LONG DISTANCE networks.

The ARBITRATION and CONTROL CIRCUITRY performs three distinct functions: First, it is responsible for formatting message requests that originate in a given PROCESSING BLOCK and forwarding these messages accordingly; second, it detects incoming messages from other blocks and determines the availability of a path that would move the incoming message closer to the destination and -- if more than one such path exists -- selects one of the available paths and forwards the message down that path, adjusting the addressing information accordingly. If a path is NOT available, this circuitry prevents the incoming message from selecting this path, forcing a different path to be established. Third, the Arbitration and Control block is also responsible for determining that an incoming message has reached its destination; if the message is at its destination, the Arbitration and Control circuitry first checks that the resource being

requested in the message header is available, and, if so, returns an ACKNOWLEDGE signal. If no acknowledgement is received by the source within a set period of time, the host will issue re-tries. After the path is established, the incoming message payload (data) is placed into the desired resource(s) by the Arbitration and Control circuitry. Each time a message attempts to establish a path through a given messenger, a state bit within the messenger's RANDOMIZATION BLOCK toggles to the opposite state. The state bit is toggled regardless of whether the attempt to establish the path was successful or not. The state bit controls which of the two output ports (at most, two output ports will move a given message closer to its destination) will be "tried" first when attempting to establish a message path. This insures that subsequent retries (should they be required) will automatically attempt to employ different paths/resources on each retry.

The RANDOMIZATION BLOCK also includes a small pseudo-random number generator implemented as a maximal length linear-feedback shift register (LFSR). The starting code for this LFSR is a function of the physical row/column information of the individual PROCESSING BLOCK with which it is associated. When a message path cannot be established or if the message destination is busy, the message source will retry the communication repeatedly, first by delaying 1, 2, 3, and then 4 clock cycles between reties. Should the communication channel still not be available after the initial try and the four, sequentially increasing clock-cycle based delays as described, the random count of the LFSR is then used to determine the number of clock cycles to delay before subsequent retries. This randomization provides a mechanism that eliminates the opportunity for deadlock situations, wherein multiple sources repeatedly attempt to communicate with a single destination or otherwise repeatedly compete for a resource. The randomization -- after the initial, aggressive 1,2,3, and 4 clock cycle retry sequence -- assures that no two subsequent retry sequences and timing will be identical. This provides intrinsic load-leveling and maximizes channel and resource availability.

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US04/037761

International filing date: 12 November 2004 (12.11.2004)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/519,129
Filing date: 12 November 2003 (12.11.2003)

Date of receipt at the International Bureau: 07 January 2005 (07.01.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)